Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №6

по дисциплине «ЭВМ и переферийные устройства»

на тему «Вывод информации на видеотерминал»

Выполнили: студенты группы 22ВВП1

Беляев Д. И.

Демин М. С.

Сергунов М. Р.

Приняли:

Никишин К. И.

Патунин Д. В.

Пенза 2025

**Цель работы**

Изучить принцип вывода алфавитно-цифровой информации на видеотерминал.

**Задание**

Задание1: Разработать и отладить программы выводом на монохромный видеотерминал, используя функцию 0 прерывания 16Н и функции 0 и 0EH прерывания 10Н BIOS; а также функции 1 прерывания 21Н DOS.

Задание2: Разработать и отладить программу вывода на экран цветного видеотерминала строки текста (имя и фамилию студента), используя функции прерывания 10Н BIOS, управляющие маркером и осуществляющие вывод символа с атрибутом. Текст должен быть выведен на фоне типа "триколор", соответствующим цветам флагов (выбрать самостоятельно)

**Ход работы**

*Задание 1*

Листинг:

stseg segment para stack

; Стек (100h слов)

dw 100h dup (?)

stseg ends

dseg segment para

; Объявление переменной для строки

msg db 'Hello, World!', 0

dseg ends

cseg segment para

lab61 proc far

assume cs:cseg, ds:dseg, ss:stseg

; Инициализация сегмента данных

mov ax, dseg

mov ds, ax

; Установка видеорежима 03h (текстовый 80x25)

mov ax, 0003h

int 10h

; Вывод сообщения через int 10h, функцию 0Eh

mov si, offset msg

print\_loop:

lodsb ; Загрузка символа из DS:SI в AL

test al, al ; Проверка на конец строки (нулевой символ)

jz wait\_key ; Если символ равен нулю, переход к ожиданию ввода

mov ah, 0Eh ; Функция BIOS для вывода символа

mov bh, 0 ; Номер страницы

int 10h

jmp print\_loop

wait\_key:

; Ожидание ввода символа (с эхом) через int 21h, функцию 01h

mov ah, 01h

int 21h

; Завершение программы

mov ax, 4C00h

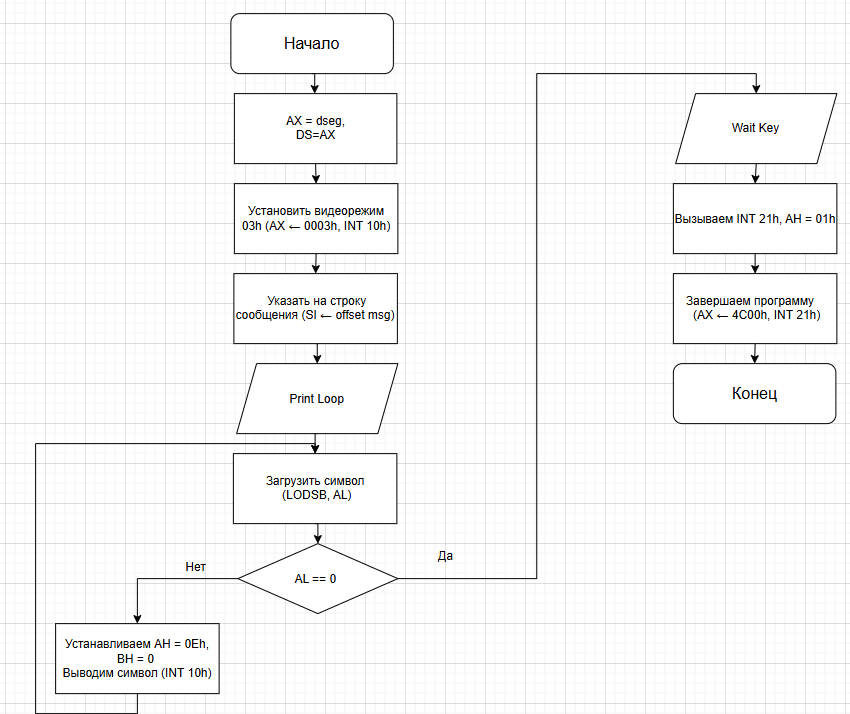
int 21h

lab61 endp

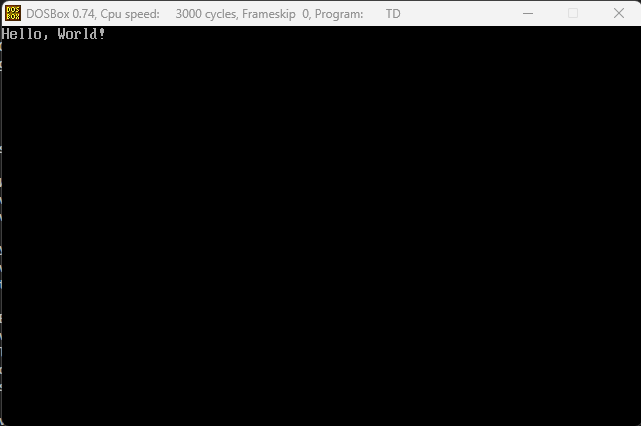
cseg ends

end lab61

Блок-схема программы:



Результат работы программы:

****

*Задание 2*

Листинг:

stseg segment para stack

; Стек (100h слов)

dw 100h dup (?)

stseg ends

dseg segment para

; Объявление переменной – строка с завершающим нулём

msg db 'Belyaev Demin Sergunov', 0

; Координаты для вывода текста

text\_row db 12 ; Номер строки

text\_col db 30 ; Номер колонки

dseg ends

cseg segment para

lab62 proc far

assume cs:cseg, ds:dseg, ss:stseg

; Инициализация сегментов данных

mov ax, dseg

mov ds, ax

mov es, ax

; Установка видеорежима 03h (текстовый 80x25)

mov ax, 0003h

int 10h

; Заполнение экрана триколором:

; Верхняя белая полоса (8 строк)

mov ax, 0608h ; Функция 06h: заполнение/прокрутка, AL = 8 строк

mov bh, 70h ; Атрибут: белый фон (7<<4 = 70h)

xor cx, cx ; Координаты: верхний левый угол (0,0)

mov dx, 074Fh ; Координаты: нижний правый угол (7,79)

int 10h

; Средняя синяя полоса (9 строк)

mov ax, 0609h ; AL = 9 строк

mov bh, 10h ; Атрибут: синий фон (1<<4 = 10h)

mov cx, 0800h ; Координаты: (8,0)

mov dx, 104Fh ; Координаты: (16,79)

int 10h

; Нижняя красная полоса (8 строк)

mov ax, 0608h ; AL = 8 строк

mov bh, 40h ; Атрибут: красный фон (4<<4 = 40h)

mov cx, 1100h ; Координаты: (17,0)

mov dx, 184Fh ; Координаты: (24,79)

int 10h

; Настройка позиции вывода текста с использованием переменных

mov dh, [text\_row] ; Номер строки

mov dl, [text\_col] ; Номер колонки

mov si, offset msg ; Указатель на строку

print\_loop:

; Установка курсора в текущую позицию

mov ah, 02h

mov bh, 0

int 10h

; Загрузка символа из строки и автоматическое увеличение SI

lodsb ; AL := [SI], SI++

test al, al

jz end\_print

; Сохраняем исходный символ (он останется в BL)

mov bl, al

; Сохраняем регистр DX, так как в нём находятся координаты курсора (DH = строка, DL = столбец)

push dx

; Чтение текущего атрибута экрана (функция 08h)

mov ah, 08h

int 10h ; После вызова: AH содержит атрибут текущей ячейки

; Сохраняем полученный атрибут в, например, BH (для вычислений не затрагивая BL, где символ)

mov bh, ah

; Восстанавливаем координаты курсора

pop dx

; Вычисление «противоположного» цвета текста:

; Извлекаем фон из атрибута (старшие 4 бита в BH)

mov al, bh

and al, 0F0h ; AL = фон в виде 0x?0

mov cl, 4

shr al, cl ; теперь AL = значение фона (число 0..F)

; Вычисляем противоположный цвет: 0Fh - фон

mov cl, 0Fh

sub cl, al ; CL теперь содержит противоположный цвет для текста

; Восстанавливаем фон из исходного атрибута

mov al, bh

and al, 0F0h ; AL = фон (высокий полубайт)

or al, cl ; AL = новый атрибут (фон + противоположный цвет текста)

; Подготовка к вызову функции 09h:

; Функция 09h ожидает: AL – символ, BL – атрибут.

; Сейчас AL содержит вычисленный атрибут, а BL – исходный символ.

xchg al, bl ; меняем местами: теперь AL = символ, BL = новый атрибут

mov ah, 09h

mov cx, 1 ; Вывод одного символа

mov bh, 0 ; Страница 0

int 10h

; Переход к следующему символу – увеличиваем столбец (DL)

inc dl

jmp print\_loop

end\_print:

; Ожидание нажатия клавиши

mov ah, 00h

int 16h

; Завершение программы

mov ax, 4C00h

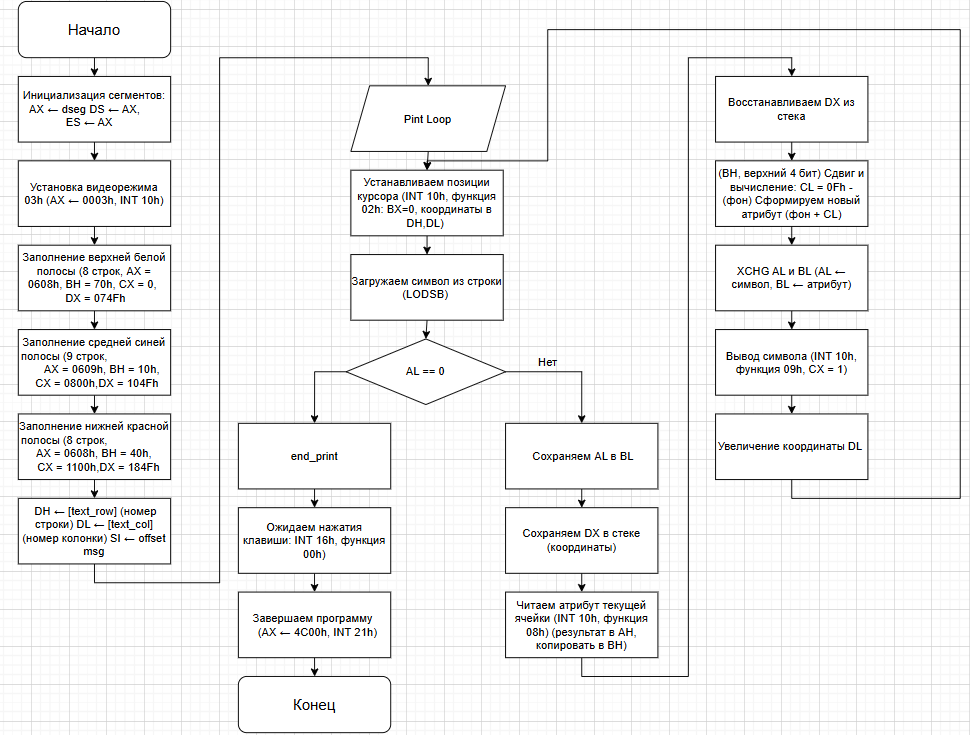
int 21h

lab62 endp

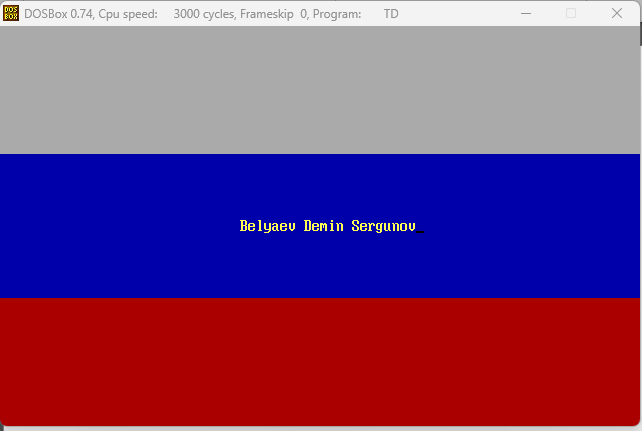
cseg ends

end lab62

Блок-схема программы:



Результат работы программы:

****

**Вывод**

Изучили принцип вывода алфавитно-цифровой информации на видеотерминал.